I claim:

1. A method for defending against a denial of service attack perpetrated by a TCP stateless hog, the method comprising the steps of:

receiving a TCP connection request from a network source and establishing a TCP connection in response thereto;

sending a keep-alive packet associated with said TCP connection to the network source, wherein said keep-alive packet comprises an invalid sequence number;

receiving a response to the keep-alive packet associated with said TCP connection from the network source; and

determining whether the network source is a TCP stateless hog based on the received response.

2. The method of claim 1 wherein the invalid sequence number is derived based on a randomly generated number.

3. The method of claim 2 wherein the invalid sequence number is derived further based on a current sequence number.

4. The method of claim 3 wherein the invalid sequence number is derived based on a formula which ensures that the invalid sequence number is numerically distant from the current sequence number.

5. The method of claim 3 wherein the received response comprises an acknowledgement number, and wherein the step of determining whether the network source is a TCP stateless hog comprises comparing the acknowledgement number to the current sequence number.

6. The method of claim 5 wherein the step of determining whether the network source is a TCP stateless hog comprises determining that the network source

is a TCP stateless hog when the acknowledgement number is not within a predetermined numerical distance from the current sequence number.

7. The method of claim 1 further comprising the step of removing said TCP connection when said step of determining whether the network source is a TCP stateless hog has determined that the network source is a TCP stateless hog.

8. The method of claim 1 further comprising the steps of maintaining a numerical count of idle connections, and removing said TCP connection when said numerical count of idle connections exceeds a predetermined idle connection limit.

9. An apparatus for defending against a denial of service attack perpetrated by a TCP stateless hog, the apparatus comprising:

means for receiving a TCP connection request from a network source and establishing a TCP connection in response thereto;

means for sending a keep-alive packet associated with said TCP connection to the network source, wherein said keep-alive packet comprises an invalid sequence number;

means for receiving a response to the keep-alive packet associated with said TCP connection from the network source; and

means for determining whether the network source is a TCP stateless hog based on the received response.

10. The apparatus of claim 9 wherein the invalid sequence number is derived based on a randomly generated number.

11. The apparatus of claim 10 wherein the invalid sequence number is derived further based on a current sequence number.

12. The apparatus of claim 11 wherein the invalid sequence number is derived based on a formula which ensures that the invalid sequence number is numerically distant from the current sequence number.

13. The apparatus of claim 11 wherein the received response comprises an acknowledgement number, and wherein the means for determining whether the network source is a TCP stateless hog comprises means for comparing the acknowledgement number to the current sequence number.

14. The apparatus of claim 13 wherein the means for determining whether the network source is a TCP stateless hog comprises means for determining that the network source is a TCP stateless hog when the acknowledgement number is not within a predetermined numerical distance from the current sequence number.

15. The apparatus of claim 9 further comprising means for removing said TCP connection when said means for determining whether the network source is a TCP stateless hog has determined that the network source is a TCP stateless hog.

16. The apparatus of claim 9 further comprising means for maintaining a numerical count of idle connections, and means for removing said TCP connection when said numerical count of idle connections exceeds a predetermined idle connection limit.